Bit-Depth Reduction And Audio Signal Classification

Mahir Jhaveri, Nirupam Roy
The University of Maryland, College Park
mjhaveri@umd.edu, niruroy@umd.edu

Abstract

In recent years, the fields of audio signal classification and speech recognition have made huge progress. Voice assistants such as Google Home and Amazon Alexa have convinced us of the potential of speech recognition and processing technologies. With the ushering era of IoT devices, tinier than ever computing devices, and an increasing need for real time processing at the edge, it has become a need of the hour to be able to make these speech recognition systems and techniques as efficient as possible while preserving their robustness. This is especially important on the smaller client-side devices which have lower memory and networking capabilities. One way to tackle these increasing constraints on the hardware is to use a more compressed version of the audio signal. Bit-depth reduction consists of limiting the total number of possible states a digital signal can be represented in and thus leads to a reduction of both storage space and network bandwidth required to process audio. In this paper, we conduct some experiments to understand the rate of decay in classification accuracy with a decrease in the quantization levels of the audio samples. Our results hint that, in most cases, it may be possible to shrink the bit depth of the audio data at the cost of very little classification/recognition accuracy.

1 Introduction

With the recent developments in IoT, it is natural to question the potential of internet-of-thing devices and audio classification. From acoustic scene analysis for threat detection to small IoT based on-body devices to continuously monitor patients based on essential bodily sounds, the possibilities of IoT with audio classification are endless.

This very naturally leads us to our next big concern, efficiency. Small IoT devices present many constraints - considerably lesser memory, computational power, and networking capacities as compared to a personal computer or a laptop. The small sizes also limit the quality of sensors, for example a microphone, which can be attached and thus lower the resolution of the data collected. We also need to think about energy efficiency when dealing with such devices, as in some cases they need to operate with limited energy supply (such as a battery). Also, computing and energy efficiency is a big concern when we are dealing with continuously running realtime audio processing systems.

One way to increase efficiency is by reducing the bit-depth [1-2] of the recorded audio sample. This essentially means reducing the number of bytes we use to represent each sample of the signal collected. This simple compression technique leads to the loss of accuracy of audio signal classification. Our hypothesis is that the loss in accuracy should not be much and we have conducted three experiments to increase our confidence in the same.

1. We use the audio clip of a simple recorded English statement and generate a transcript for it using IBM Watson. We then repeat this experiment for the same audio clip reduced to different bit depths. We compare this generated transcript to the original transcript using a metric called Word Error Rate and then compare the accuracies for different bit-depths.
2. We build a simple audio classifier using a convolutional neural network(CNN) [3]. We then train and test the model for identical samples at different bit-depths and then compare the accuracies achieved. Each sample is an audio recording of a single instrument playing and the task is to identify which instrument it is.
3. Similar to experiment 2, we build another audio classifier using CNNs. But, this time we use the dataset from another domain. Each sample is an audio recording of a person's heartbeat and the task is to classify the heartbeat as normal or abnormal.

2 Background

In this section, we talk about some of the concepts and metrics we have used in designing our experiments. Here, we first talk about the concept of bit-depth (quantization levels of a signal) and analog-to-digital conversion. Then, we move into exploring the ideas of the Word Error Rate metric, Convolutional Neural Networks.

2.1 Bit-depth Overview

Sound signals travel in the form of pressure changes and they are analog in nature, that is they vary continuously. But, in order to store them digitally, we need to convert them to a discrete set of values. This is done by sampling

the signal at a certain frequency known as the sampling rate. And according to a well-known theorem in information theory, known as the Shannon Nyquist Sampling theorem[1-2], an analog signal can be losslessly converted to a digital signal if the sampling rate is twice the maximum frequency contained in the signal. If f is the maximum frequency present in the signal (whose information we chose to retain) then, we must choose a sampling frequency fs such that,

$$fs = f/2$$

to ensure no loss of information in that band limit. Since the signal is analog in nature, it can take up infinitely many values or states, but it is not possible to represent all of it in memory. So, we represent this signal in memory by mapping these continuous, infinite values to a smaller set of discrete values. A process called Quantization. The number of possible discrete values the signal can take up is determined by the number of bits we use to represent each sample, this is called the bit-depth of the signal. If b is the bit-depth of the signal then,

$$number\ of\ states = 2^{b}$$

Quantization decreases the information content of the signal and produces some error known as Quantization error[1-2]. But bit-depth reduction can greatly reduce the memory requirements for signals and improve the efficiency of audio classification systems. For example, reducing bit-depth from 16 bits per sample to 8 bits per sample reduces the memory requirement to 50% which is substantial when dealing with continuous, realtime models. Our experiments analyze the classification accuracy when using reduced bit-depth samples.

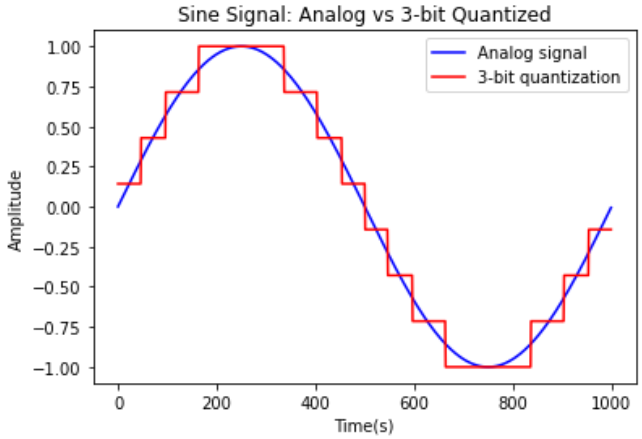

Figure 1

Figure 1, shows the original sine curve in blue and the sine curve with a quantization level (ie. bit-depth) of 3 bits. Here, we should not that although in memory we are still using 16 bytes to stores each level we are confining the signal to take one of the 9 different values for each time instance and thus simulating the effect of reducing the bit-depth(ie. The classifier is given only as much information as it would get from a signal with a 'real' quantization level of 3 bits).

Algorithm 1 Reduce_Bit_Depth
Input: signal, new_bit_depth
Output: new_signal
1: **Scale** signal to fit between [-1, 1]
2: bins = [2^(new_bit_depth) equally spaced values in the range [-1,1]]
3: new_signal = []
4: **for** each t in signa.size():
5: new_signal[t] = **round** signal[t] to closest value in bins
6: **return** new_signal

Figure 2

Figure 2, shows our algorithm for reducing the bit-depth of a given signal.

2.2 Word Error Rate (WER)

Word Error Rate is a metric used to measure the similarity between two sentences. We use WER to compute the error in transcribing a given audio sample. The metric is calculated by first aligning the two strings using a bioinformatics algorithm called dynamic string alignment and then using the formula,

$$WER = \frac{S + D + I}{N}$$

where S is the number of substitutions, D is the number of deletions, I is the number of insertions and N is the number of words in the reference string.

2.3 Convolutional Neural Networks

For two of our experiments, we attempted to build our own simple audio classifier, and in this section, we attempted to provide a brief overview of the same. Deep neural networks have become a popular and successful mechanism for solving supervised learning problems today [3]. So we decided to use neural networks for building our classifiers. Now, the classification of audio signals has two important steps: first is the preprocessing of audio signals into discrete features suitable for learning, and second is building the model, which is choosing the architecture, optimizers, loss function, and so on. Now, the time-domain representation of sound waves is not the best to use for deep learning based models. Instead what we do is we first convert it to the time domain and compute something known as the Mel-frequency cepstral coefficients (mfcc) for windows of the signal at regular intervals of time. Now, when the architecture of the actual model is concerned we have chosen something known as a convolutional neural network (CNN). Now although CNNs are traditionally used for dealing with image classification problems they, in general, work well when data to be classified have spatial relationships. We chose CNN over Recurrent Nets, their counterparts traditionally used on audio datasets, due to their speed and accuracy on our small dataset.

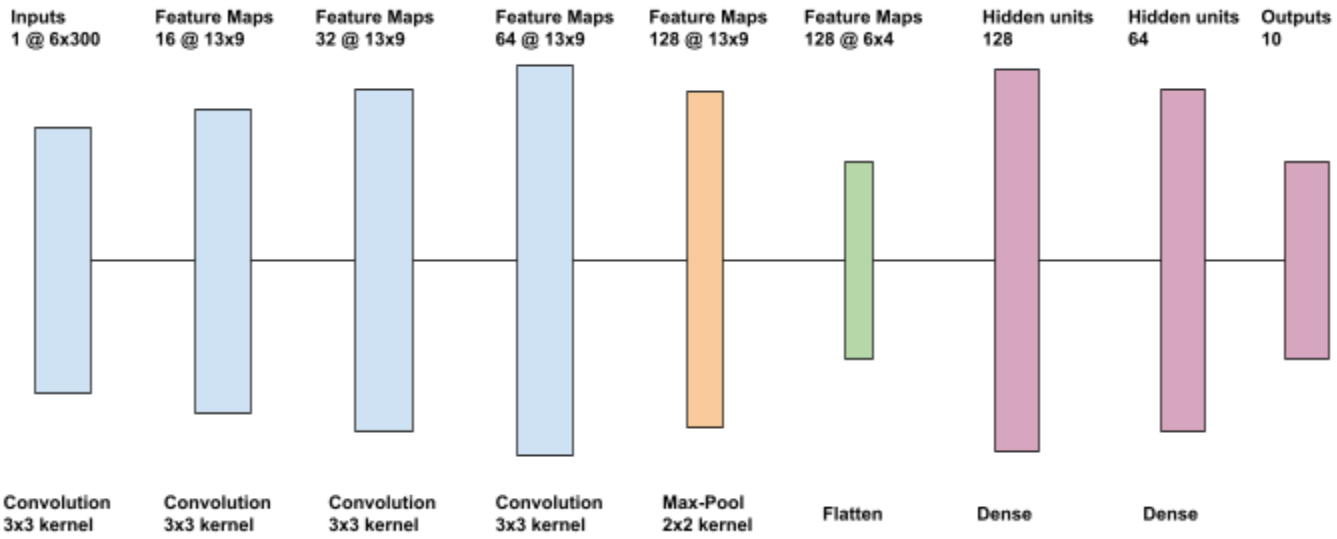

Figure 3

3 Experiments

In this section, we discuss some of the experiments we conducted in detail.

3.1 Prediction accuracy of IBM Watson on reduced bit-depth samples

The goal of this experiment is to show the variation of the prediction accuracy with reduction of bit-depth. First, we simply recorded a short clip of a person saying the sentence "the major goal of introduction to compilers is to arm students with the ability to design, implement and extend a programming language". The original signal had a 16 bit-depth. We then reduced the bit-depth of the signal to 15, 14, and so on, down to 2. We then used the IBM Watson, which is a state of the art speech recognition engine, API to fetch the transcript for these samples. Next, we used the word error rate metric to measure the closeness of these generated transcripts to the original transcript. We repeat this procedure on the same samples after adding some white Gaussian noise, to see the classification accuracy when noise is present.

3.2 Audio classification of reduced-bit depth samples of instruments data

In this experiment, we attempt to build our own supervised learning based CNN classifier to identify sounds of different musical instruments and compare the prediction accuracies for different bit-depth levels. The goal of this experiment was to see if a simple CNN based network can learn to differentiate between sounds having lower bit-depths. For this experiment, we used an open-source audio dataset available on Kaggle as a part of the Freesound General-Purpose Audio Tagging Challenge [4]. Each sample in the dataset consists of audio recordings of a single instrument playing. The instruments in the dataset are - Acoustic Guitar, Bass Drum, Celo, Clarinet, Double Bass, Flute, Hi-hat, Saxophone, Snare Drum, and Violin. The task is to correctly identify the instrument in each of the samples. 80% of the available data was used for training while the remaining formed the test set.

The first step involved in this experiment was converting the time-domain representation of the signals into frequency-domain. For this, we chose to compute the Mel-Frequency Cepstral Coefficients for the signal. The number of mfcc to return, window size, hop length, etc. were treated as hyperparameters and tuned till we achieved the desired accuracies.

Next, this mfcc data is fed into a CNN model built using Keras. The model consisted of 4 convolutional layers one after another followed by a max-pooling layer and then followed by 3 dense layers. Figure 3 provides more details about the layers. We used relu as our activation function on all layers besides the final dense layer which had softmax. The loss function used for training was categorical cross entropy and the optimizer used was Adam. We trained and tested our model for different quantization levels of our dataset and evaluated the results.

3.3 Audio classification of reduced-bit depth samples of heartbeat data

This experiment is similar to the last experiment with three main differences.

One, the dataset we use is a part of The PhysioNet Computing in Cardiology Challenge 2016 [5]. The dataset consists of heartbeat recordings samples labeled as either normal or abnormal. The challenge is to correctly label the samples. Again, we use 80% percent of the samples for training and the remaining 20% for testing.

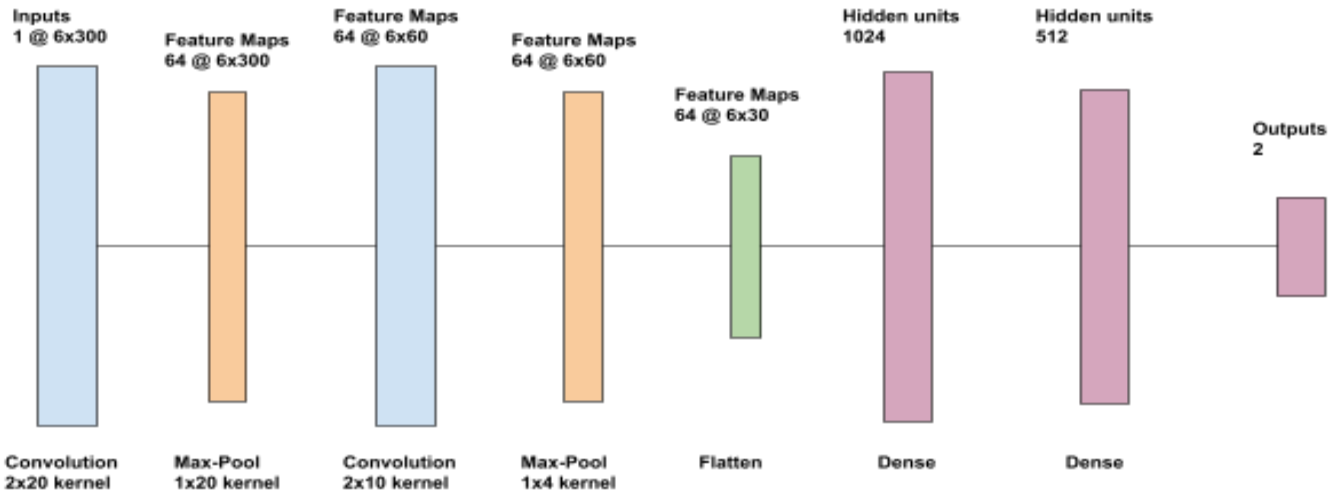

Figure 4

Second, this experiment involves a small extra pre-processing step. Before reducing the bit-depths, we use something called Springer's Algorithm [6], a Hidden semi-Markov Model based heart sound segmentation algorithm, to align all our samples to start at the first heart sound. This algorithm is provided to us along with the dataset. This step is recommended by the Physionet challenge as this helps increase classification accuracy. All bit-depth reductions are made after this step. The impact of reducing bit-depths on Springer's Algorithm is not tested as a part of this paper.

Finally, the architecture for the CNN model we use is based on the paper published by Rubin et al. as a part of their solution to the challenge [7]. The model consists of 2 convolution layers each followed by a max-pool layer. This is followed by a flatten layer and 2 dense layers. Figure 4 provides more details about the layers used. The activations we use are ReLU, for all units except for the output layer, and Sigmoid in the output layer. The loss function is Binary Crossentropy and the optimizer is Adam.

4 Evaluations

In this section we go over our observations and analysis of the experimental results.

4.1 Prediction accuracy of IBM Watson on reduced bit-depth samples

In the first experiment when we tried to classify the sound samples (no noise) of different bit-depths to test the accuracy. Figure 5 records our observations.

As it can be seen, in the no noise samples, the word error rate is very low, in fact zero, for samples with bit-depth greater than or equal to 7. The bit-depths of 6 and 5 have small word-error-rate of around 4 and the error blows up for samples with bit-depths lesser than or equal to 4. These results give us some hope that low bit-depth audio can be used for speech recognition in low noise environments.

Next, we ran the speech recognition api on noisy data, created by using the exact same 15 samples by adding white gaussian noise to them. Here, the amount of noise is measured by a metric called sound to noise ratio (abbreviated SNR). This metric is basically inversely proportional to the amount of noise in the sample. Some of the observed results are shown in Figures 6, 7 and 8:

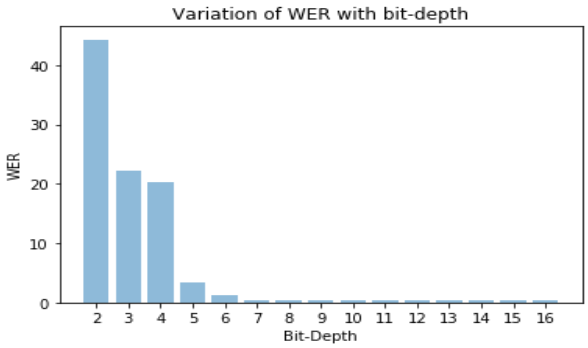

Figure 5

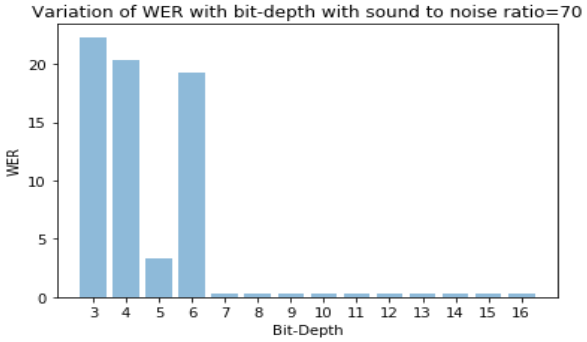

Figure 6

As it can be seen, for higher bit-depths (above 6), there is a strict increase in the inaccuracy as we add noise to the

signal. But, at lower bit-depths there is no particular order of increase or decrease. It can also be seen that in samples with bit-depth greater than or equal to 6, the error rate is very less until the much noise is added.

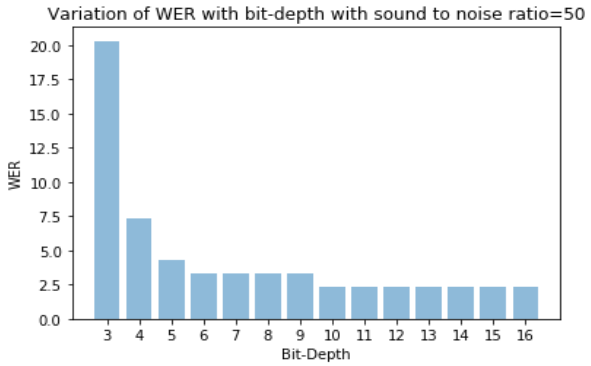

Figure 7

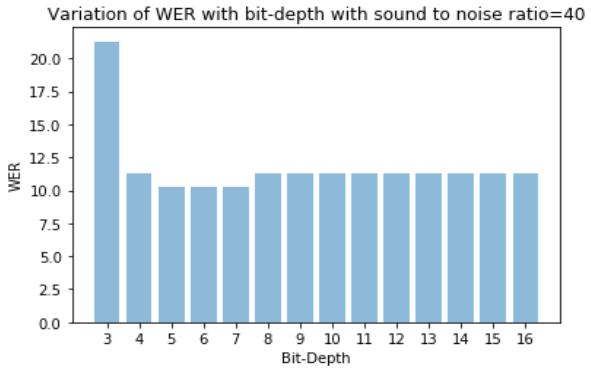

Figure 8

4.2 Audio classification of reduced-bit depth samples of instruments data

In our second experiment we had built a simple convolutional neural network based audio classifier to compare the prediction accuracies between audio samples of different bit-depths. Here, we present two charts: the final training accuracy (figure 9) with bit-depths and final test accuracy with bit-depths (figure 10).

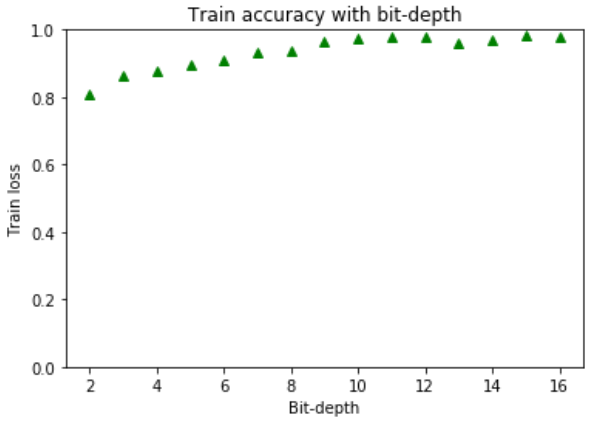

Figure 9

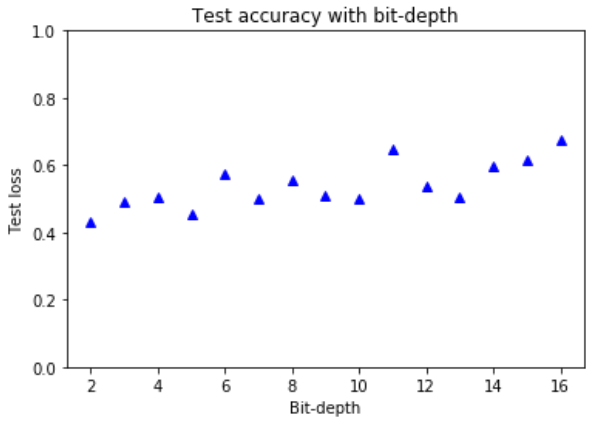

Figure 10

In these charts, it can clearly be seen that the accuracy of prediction decreases with a decrease in bit-depth. Since changing bit-depths from 16-bits to 2-bits reduces the signal's memory requirement by 8-times, the drop in accuracy that results with the change is not as drastic.

4.3 Audio classification of reduced-bit depth samples of heartbeat data

In our final experiment, we use a CNN classifier to categorize the heartbeat data as abnormal and normal for various bit-depths. The results are summarized in figure 11.

Quantization	Train Accuracy	Test Accuracy
Original (16-bit)	95.91%	89.35%
8-bit	96.07%	89.04%
7-bit	96.81%	88.27%
6-bit	95.88%	89.19%
5-bit	97.94%	85.65%
4-bit	94.82%	83.02%
3-bit	90.49%	84.88%
2-bit	88.69%	79.63%
1-bit	89.56%	82.87%

Figure 11

As shown in figure 9 we can see that both the train and test accuracies decrease with a decrease in bit-depth. But, the drop in both train and test accuracies are not that significant for either the train or test sets. This implies that reducing bit-depths does not have a significant impact on this particular dataset.

5 Ongoing and Future Work

In our ongoing work, we are building a small on-body device which has a stethoscope microphone to record heart sounds and a small chip to run simple deep neural networks. The goal is to be able to detect different bodily events such as talking, heavy breathing, coughing, eating, etc. from just an audio cardiogram recorded close to the chest. In this work, we are trying to apply ideas of bit-depth and sampling rate reductions to make the device more compact and low-power.

In this paper, we test the impact of reducing audio bit-depths on classification accuracies in 3 different domains. So, one of the most logical next steps would be to extend this work to test the impact on more audio related domains as well as not just restrict the experiments to classification but test performance on other forms of audio processing tasks as well.

Another area of future work is to not just restrict ourselves to using only bit-depth reduction. But also test variations in classification accuracies by applying another form of lowering resolution based audio compression techniques - down sampling.

Ideas of lowering resolution such as bit-depth reduction or downsampling can be studied in other domains of signal processing such as image processing and video processing.

6 Conclusion

From our experiments it can be seen that when there is negligible noise in audio, we can get away with using low bit-depth audio. In fact, it appears that audio with 5 or 6 bit-depths have almost negligible errors. Lower bit-depth audio also performs decently when the noise present in the sound is low. This idea can be incorporated into many smaller sound processing devices with low memory and power. Although this concept might need some domain specific testing, to make sure it works for the problem at hand, it can reduce memory requirements greatly. In fact, a conversion of an audio sample of 16 bit-depth to 8 bit-depth halves the memory requirement.

References

[1] R. G. Lyons, Understanding Digital Signal Processing, 1997.

[2] Steven W. Smith, The Scientist and Engineer's Guide to Digital Signal Processing, 1997.

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. 2016. MIT Press.

[4] Eduardo Fonseca, Manoj Plakal, Frederic Font, Daniel P. W. Ellis, Xavier Favory, Jordi Pons, Xavier Serra. General-purpose Tagging of Freesound Audio with AudioSet Labels: Task Description, Dataset, and Baseline. Submitted to DCASE2018 Workshop, 2018. URL: https://arxiv.org/abs/1807.09902

[5] Liu C, Springer D, Li Q, Moody B, Juan RA, Chorro FJ, Castells F, Roig JM, Silva I, Johnson AE, Syed Z, Schmidt SE, Papadaniil CD, Hadjileontiadis L, Naseri H, Moukadem A, Dieterlen A, Brandt C, Tang H, Samieinasab M, Samieinasab MR, Sameni R, Mark RG, Clifford GD. An open access database for the evaluation

[6] Springer DB, Tarassenko L, Clifford GD. Logistic regression-hsmm-based heart sound segmentation. IEEE Transactions on Biomedical Engineering 2016;63(4):822– 832.

[7] J. Rubin, R. Abreu, A. Ganguli, S. Nelaturi, I. Matei and K. Sricharan, "Classifying heart sound recordings using deep convolutional neural networks and mel-frequency cepstral coefficients," 2016 Computing in Cardiology Conference (CinC), 2016, pp. 813-816.

Appendix

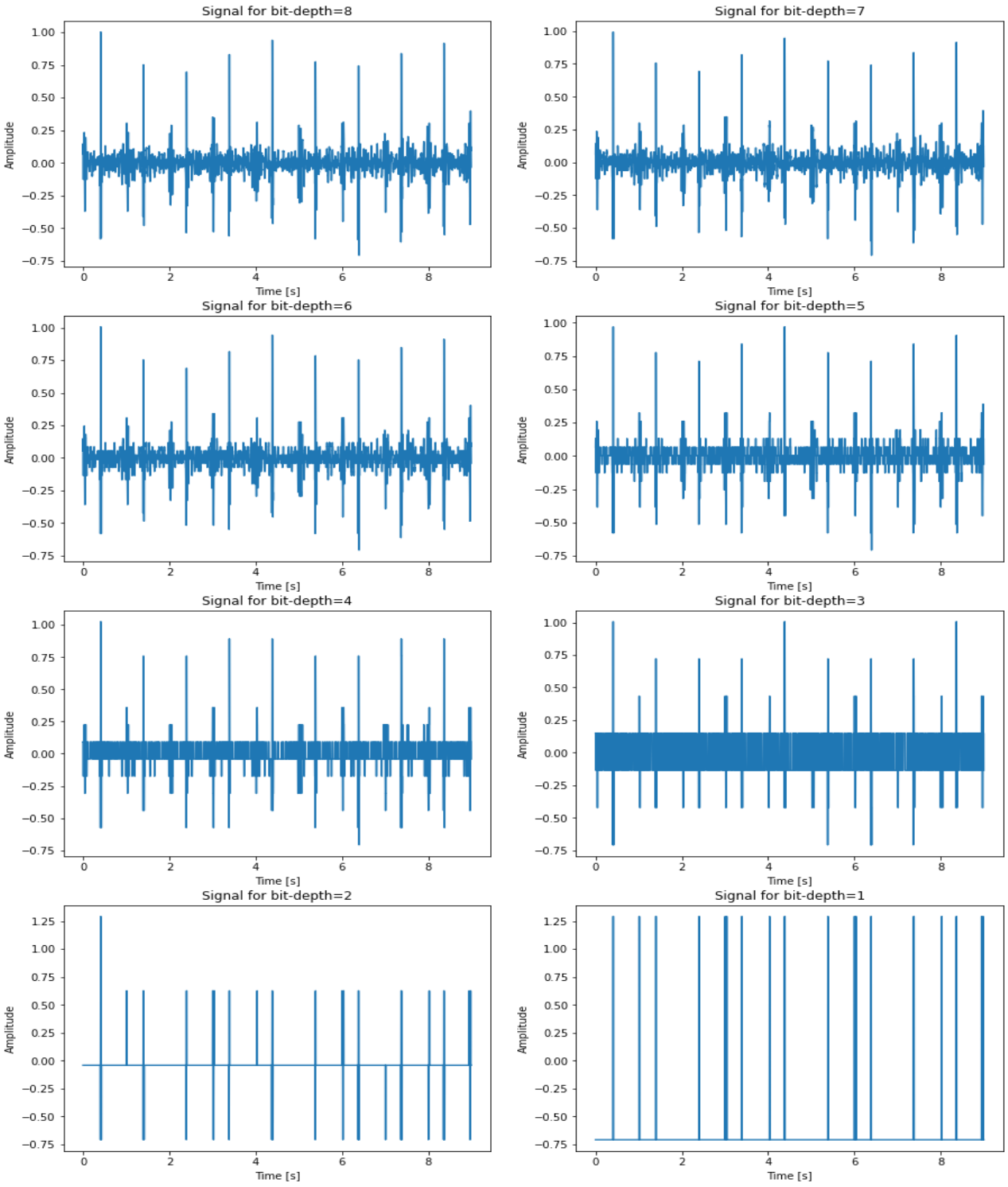

Figure 12 - The time-series charts for a signal (heartbeat dataset) at different bit-depths.

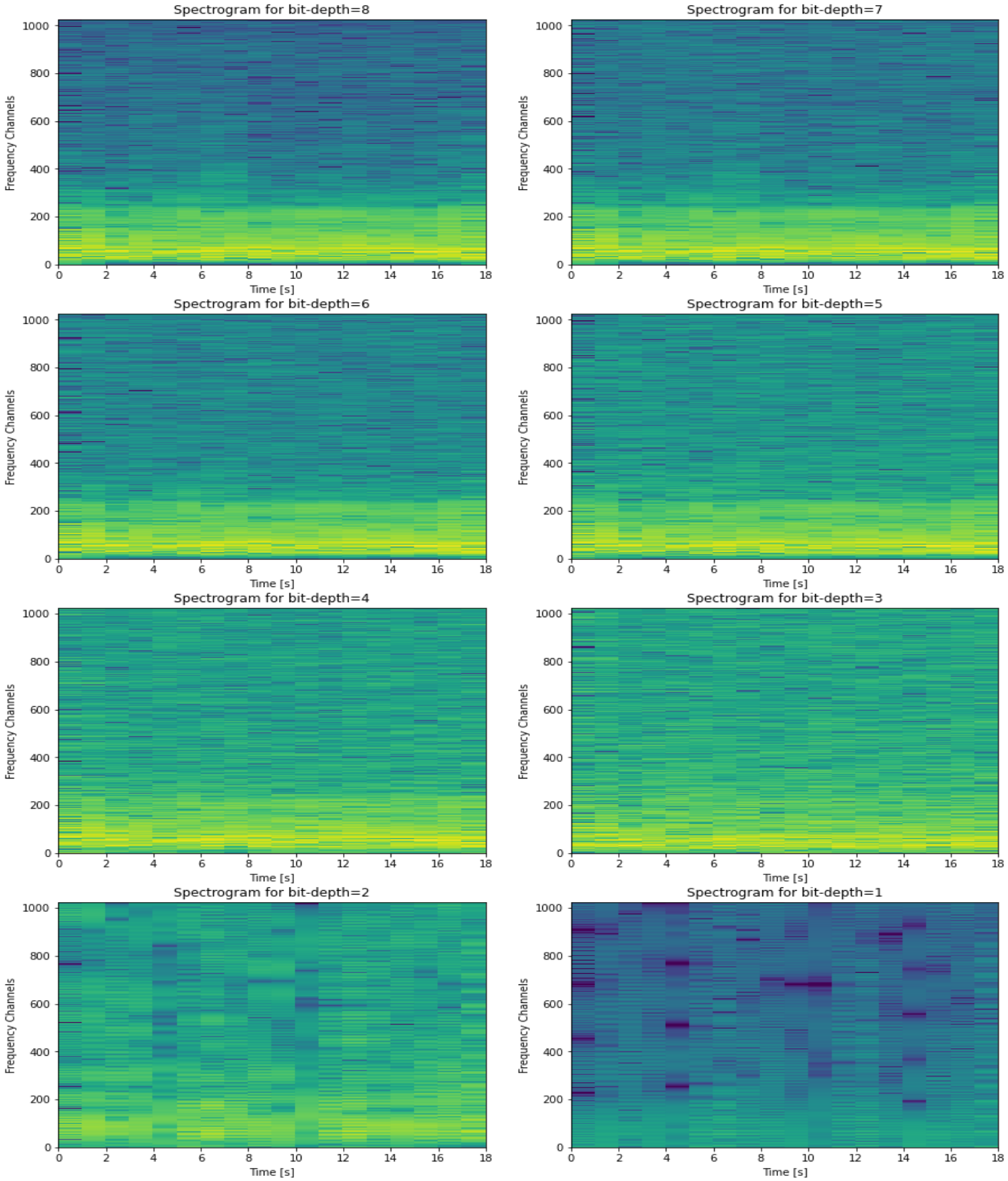

Figure 13 - Spectrograms for the same signal (as Figure 12) at different bit-depth.